National Textile University

Department of Computer Science

Subject:

Operating System

Submitted to:

Sir Nasir Mehmood

Submitted by:

Eman Babar

Reg. number:

23-NTU-CS-FL-1148

Semester:

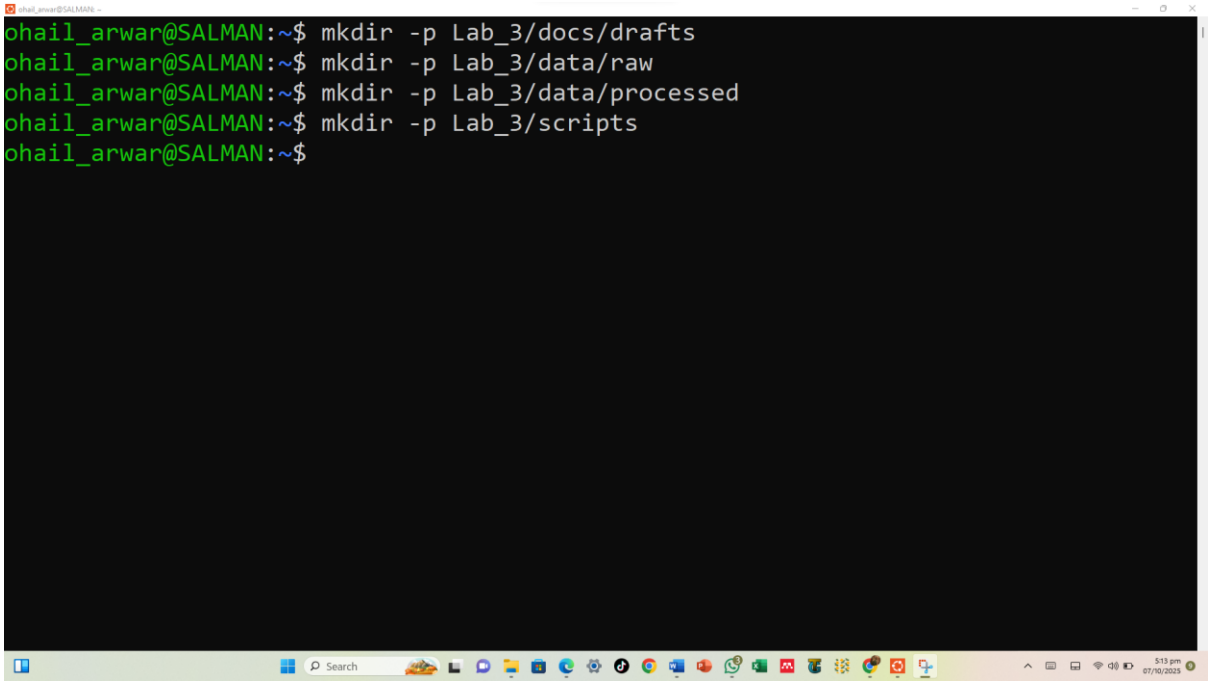
5^{th}- A

LAB-3-Home-Task

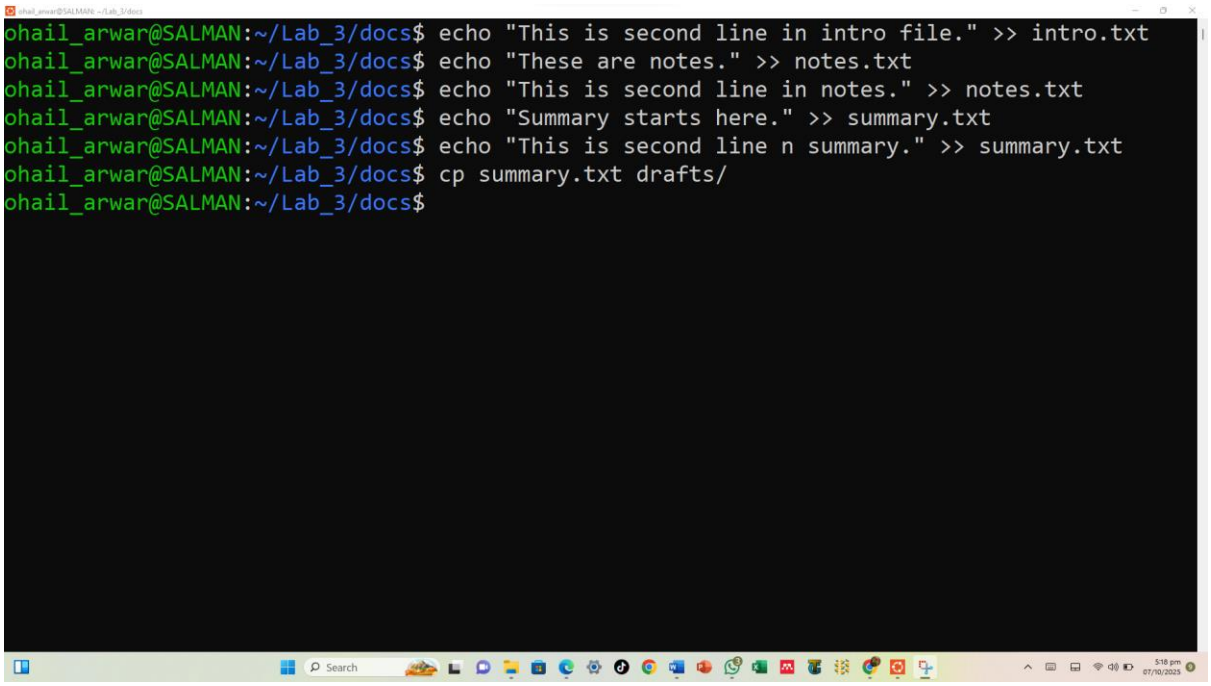
Part 1: File and Directory Operations

1. Create the following directory structure in your home directory:

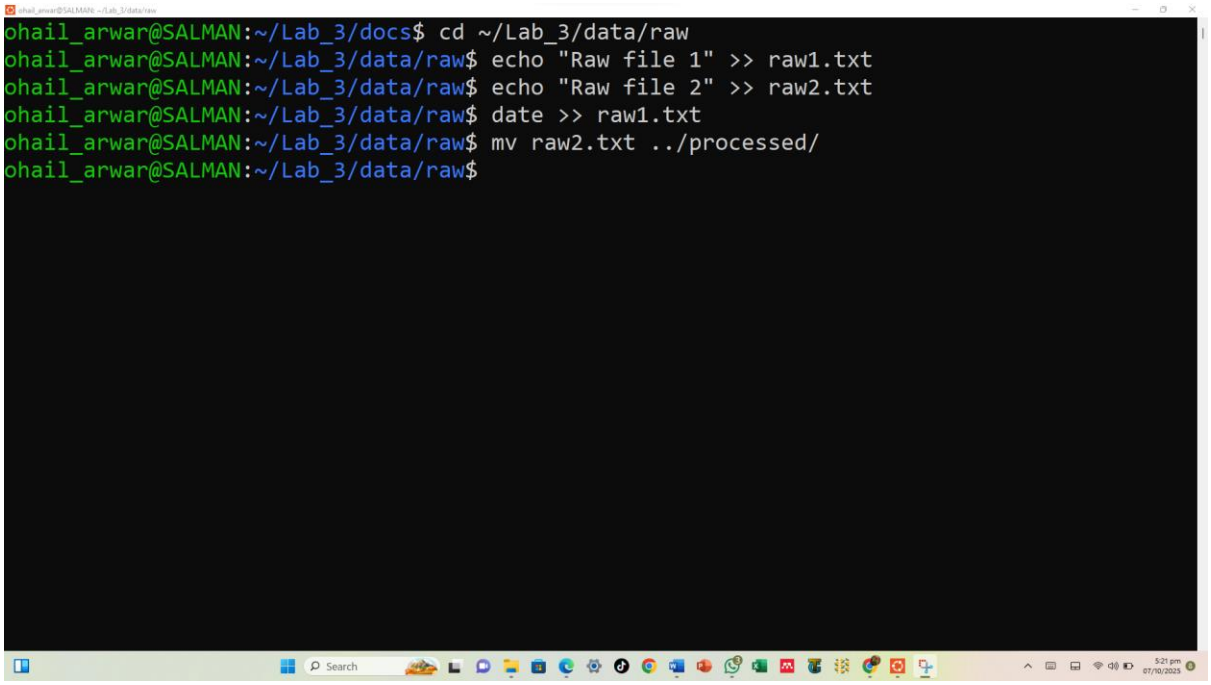
```
Lab_3/
├── docs/
│   └── drafts/
├── data/
│   ├── raw/
│   └── processed/
└── scripts/
```



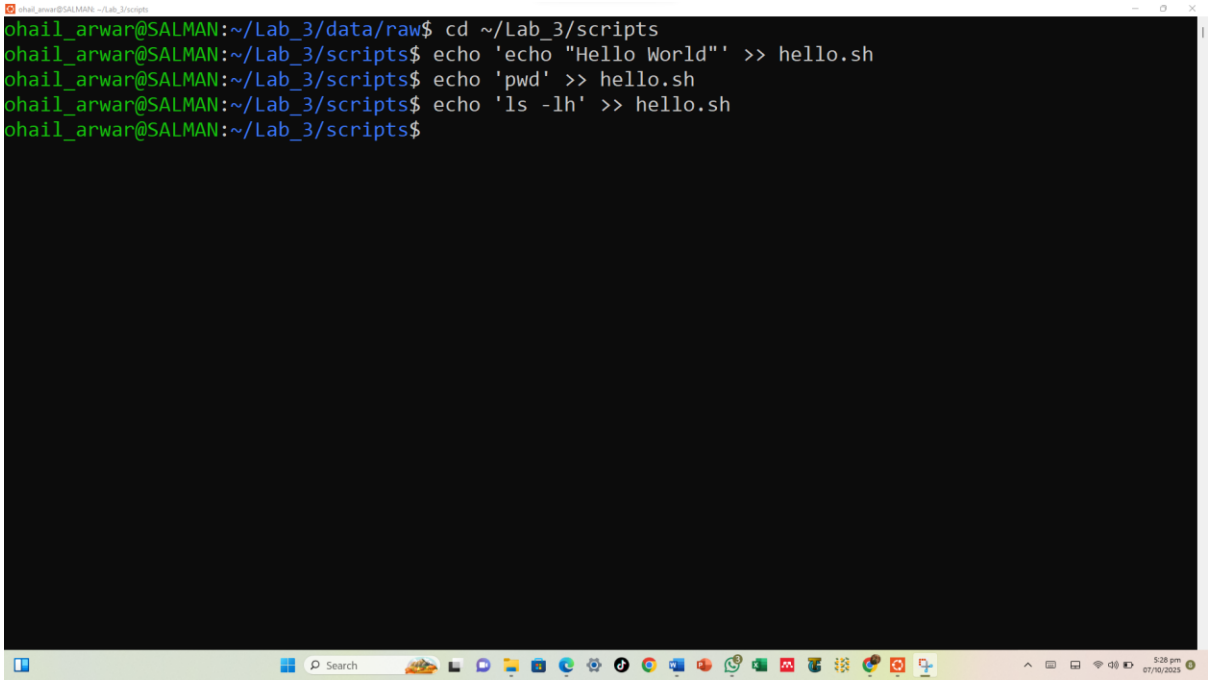
2. Inside docs/:
    - Create three files: intro.txt and notes.txt and summary.txt
    - Add at least two lines of text into each using echo >>.
    - Copy summary.txt into the drafts/ folder using **cp** command.



3. Inside data/raw/:
    - Create two files: raw1.txt, raw2.txt
    - Append the current date into cp command. raw1.txt using the date command.
    - Move raw2.txt into processed/ using mv source destination mv. The syntax is: mv source destination

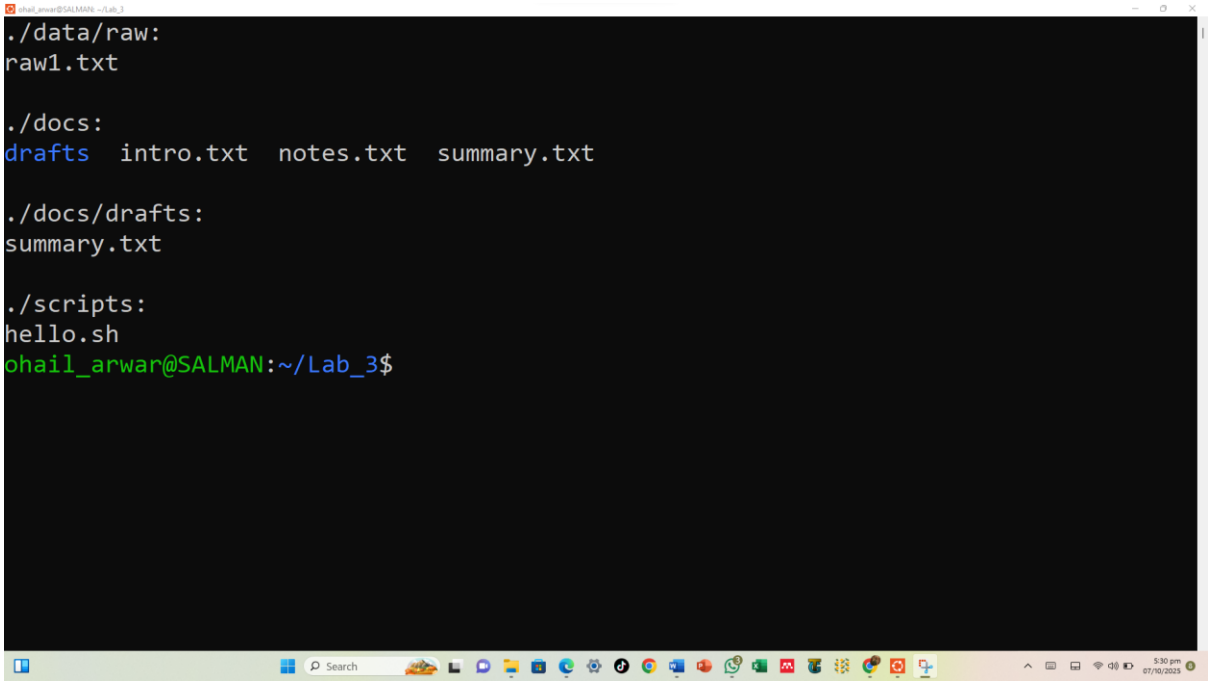


4. Inside scripts/:
    - Create a script named **hello.sh** with the following content:
      echo "Hello World"
      pwd
      ls -lh
    - Later, you will make it executable (in Part 3).



5. Display the directory structure recursively and take a screenshot:

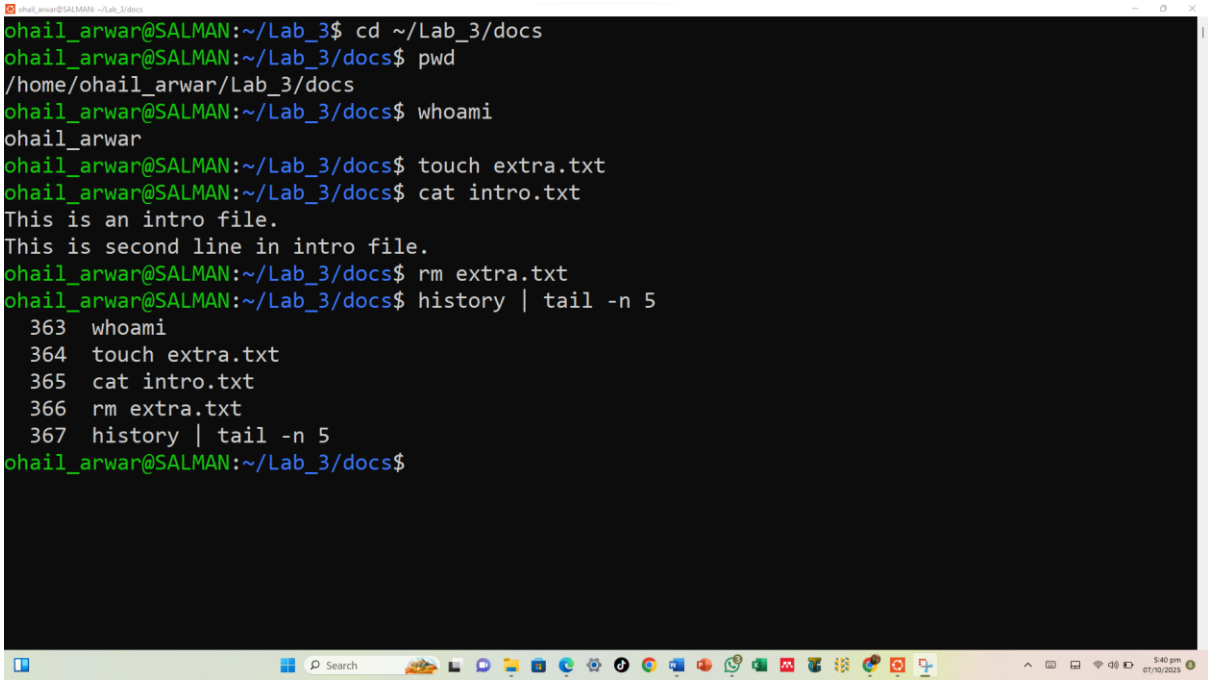
Ls -R



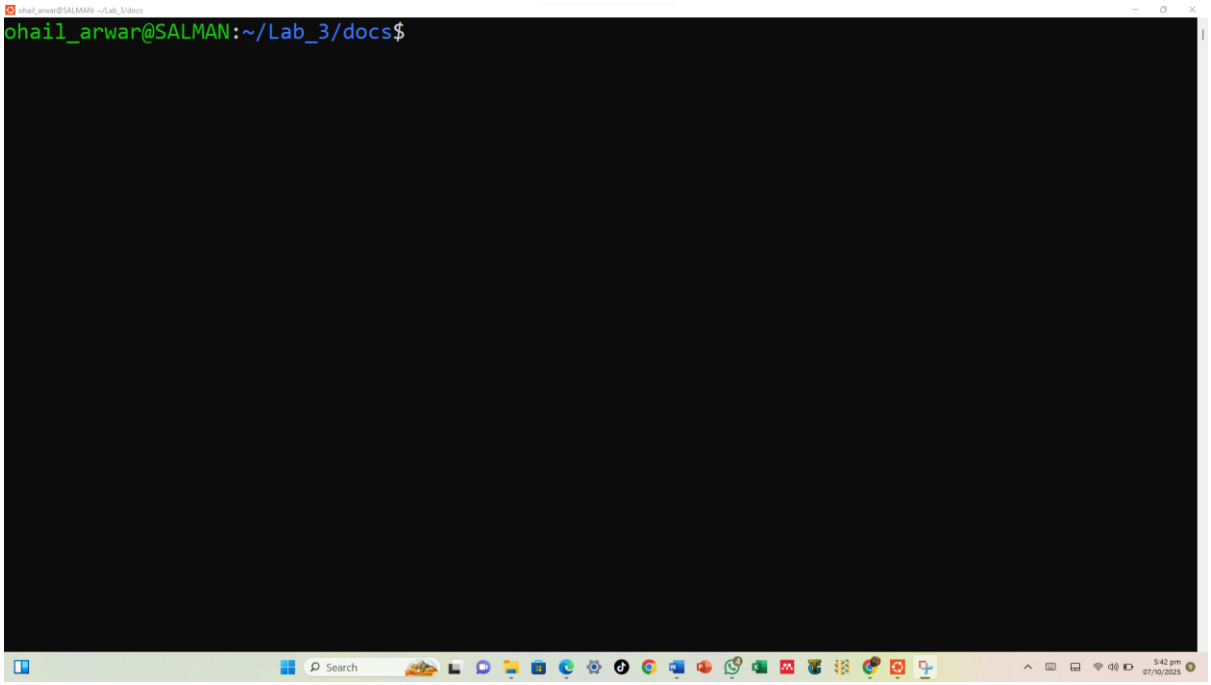
Part 2: Practice with Basic Linux Commands

Run the following commands inside Lab_3/ and note their outputs:

- pwd → Show current working directory.
- whoami → Display the current logged-in user.
- touch extra.txt → Create an empty file.
- cat intro.txt → Display file contents.
- rm extra.txt → Delete a file.
- history | tail -n 5 → Show your last 5 executed commands.



- clear → Clear the terminal. Take screenshots of commands and outputs



Part 3: File Permissions and Ownership

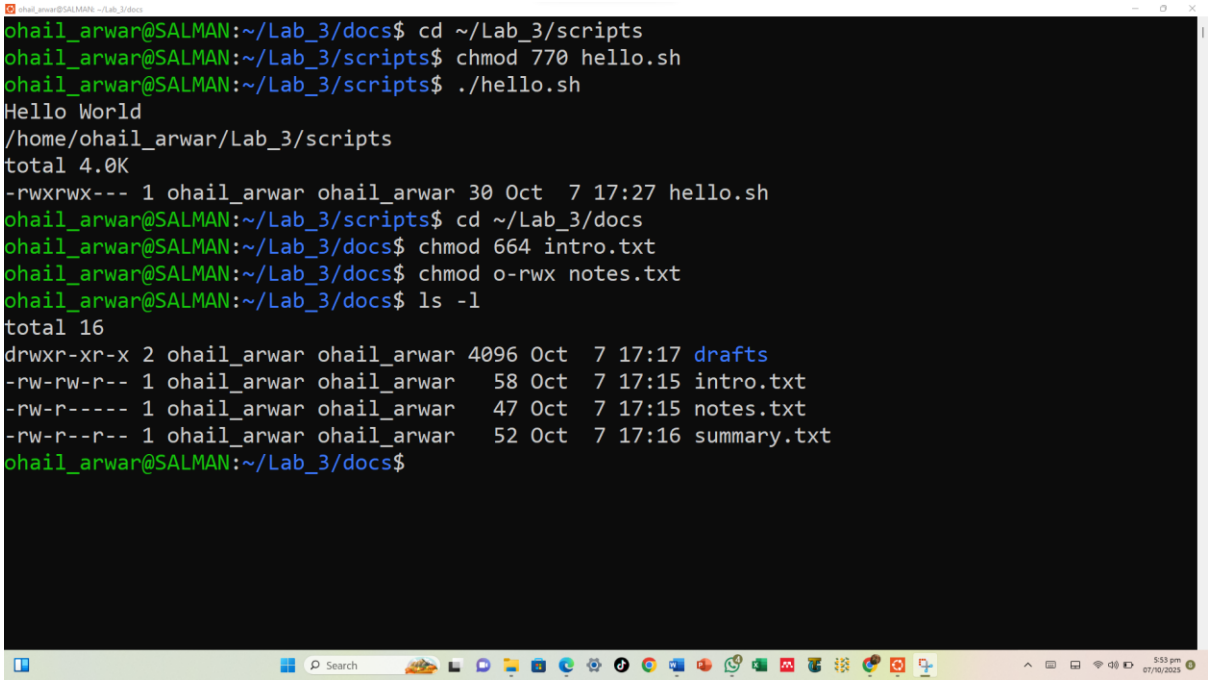
1. Change the permissions of hello.sh so that:
   - Owner → Read, Write & Execute
   - Group → Read, Write & Execute
   - Others → No permissions
   - Run the script using:

     ./hello.sh
2. Change the permissions of intro.txt using numeric notation so that:
   - Owner → Read & Write

- Group → Read & Write
- Others → Read only 3.

3. Change the permissions of **notes.txt** using **symbolic notations** so that **others** have any permission on it.
4. Verify all changes with:

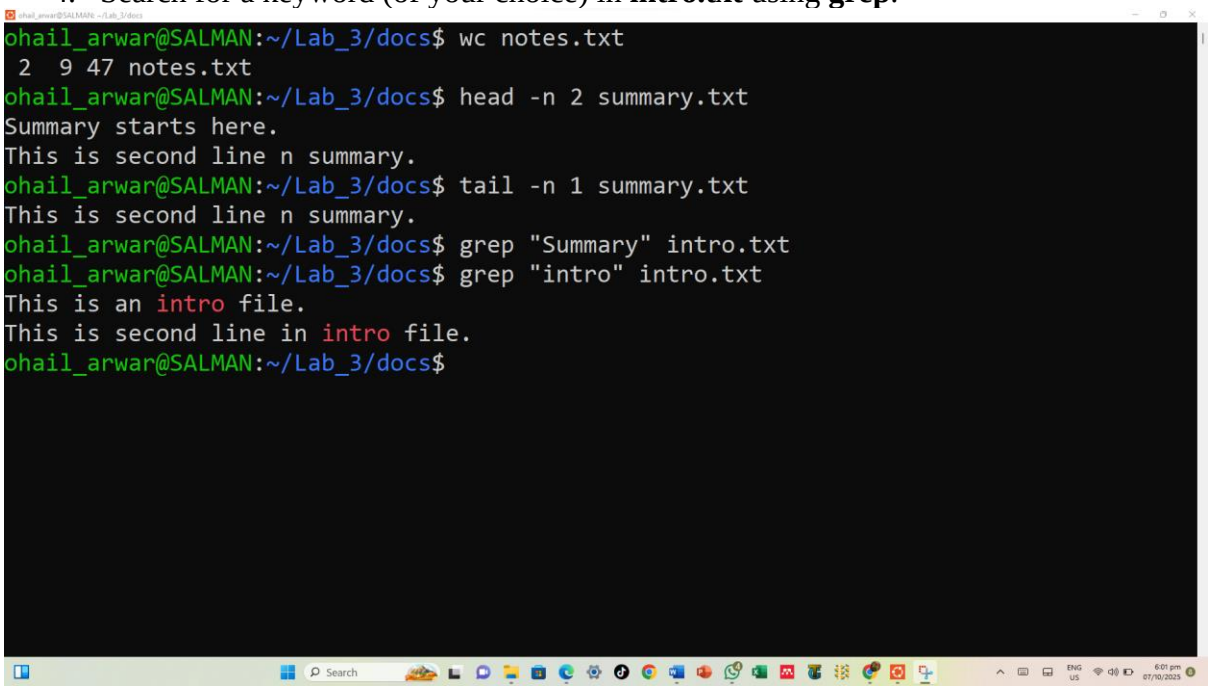
ls-l



Part 4: Reading and Searching Files

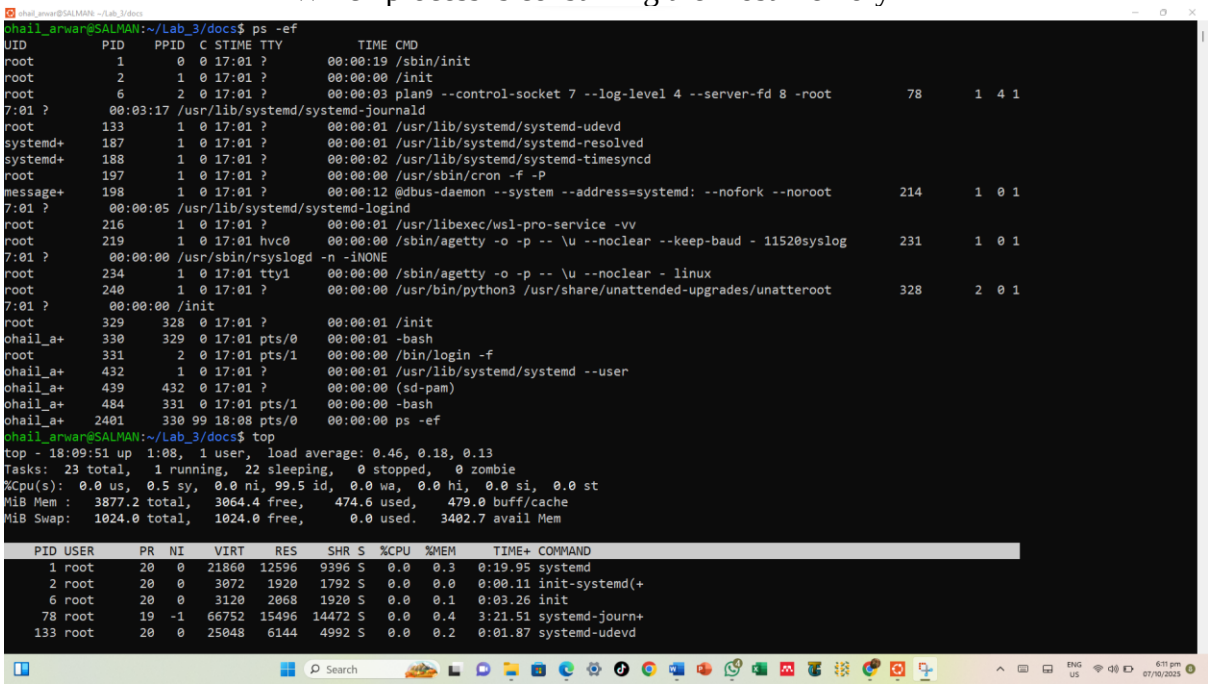
Inside docs/:

1. Count the number of lines, words, and characters in **notes.txt** using **wc**
2. Show only the first 2 lines of **summary.txt** using **head -n 2**
3. Show the last line of **summary.txt** using **tail -n 1**
4. Search for a keyword (of your choice) in **intro.txt** using **grep**.

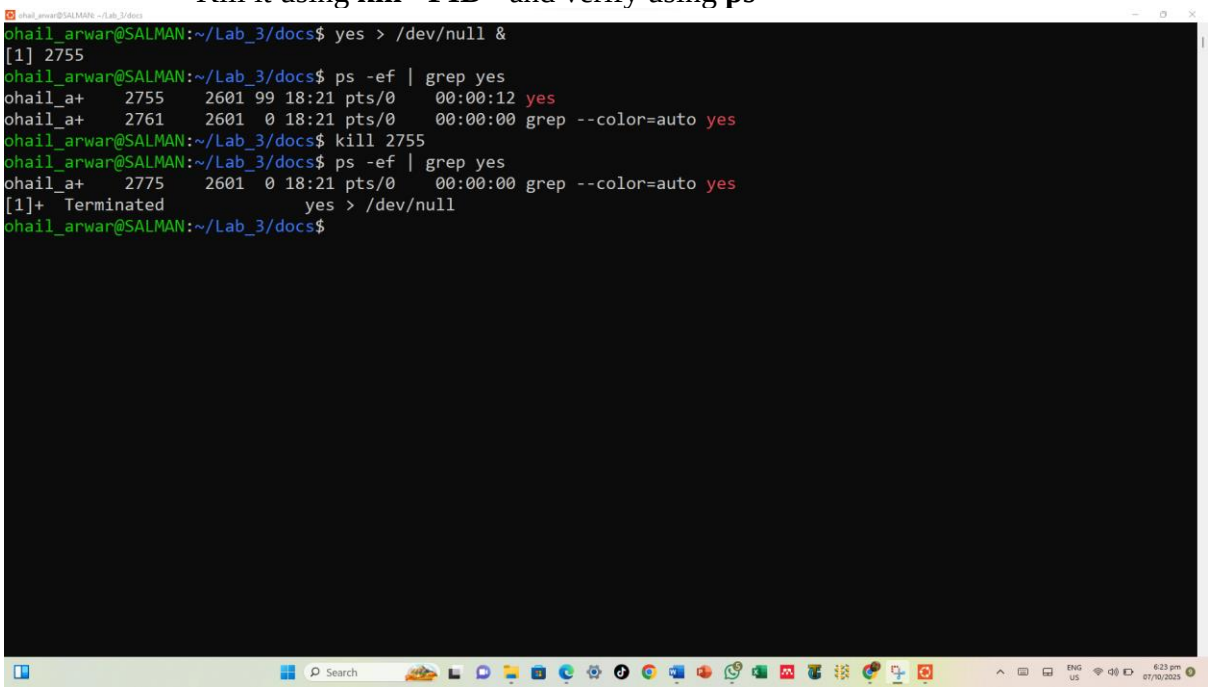


Part 5: Linux Process Commands

1. Exploring Processes
    - Use **ps -ef** and identify 3 processes running on your system. Note their PID, PPID, and command.
    - Run **top** for 20–30 seconds. Write down:
        - Which process is consuming the most CPU
        - Which process is consuming the most memory

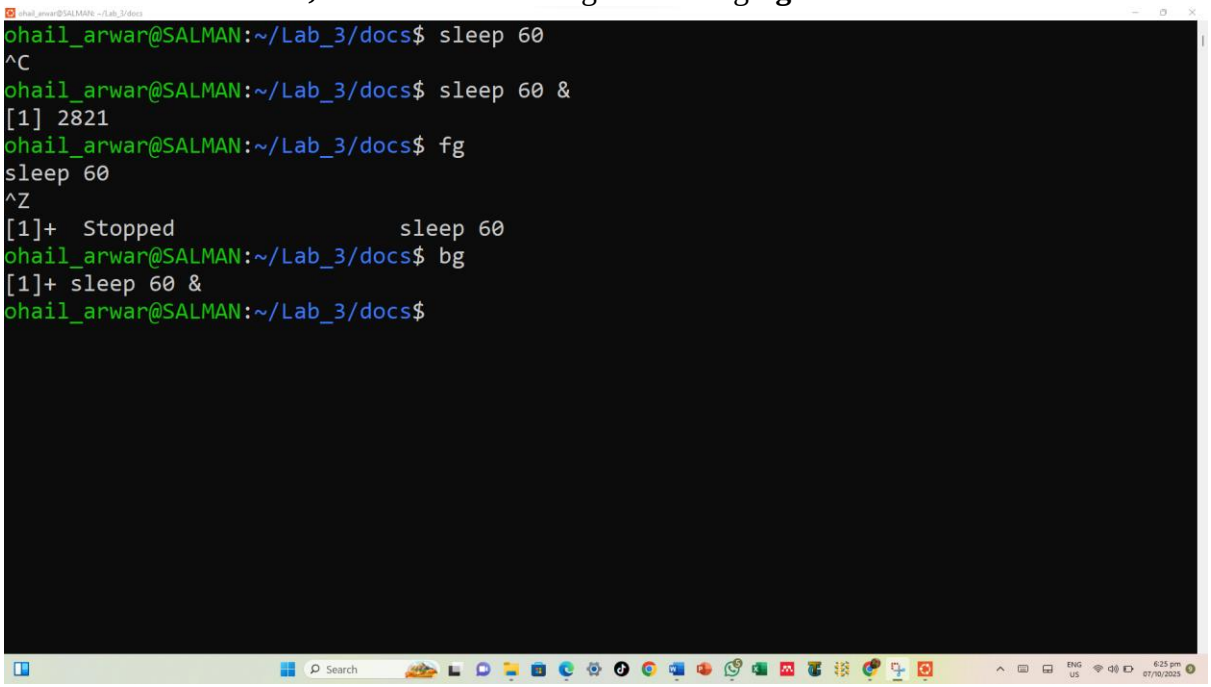


2. Practice with Infinite Process
    - Start: yes > /dev/null &
    - Locate its PID using ps -ef | grep yes
    - Kill it using **kill <PID>** and verify using **ps**



3. Foreground & Background Jobs

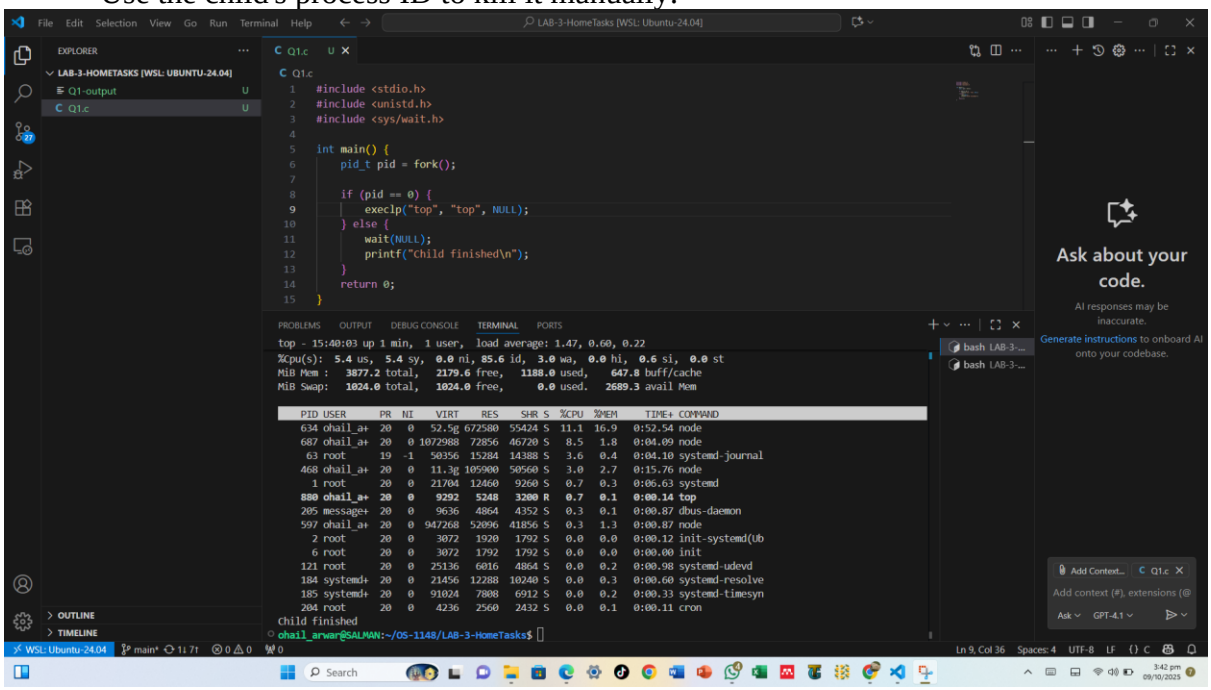
- Run **sleep 60** in **foreground** and terminate it with **Ctrl + C**.
- Run **sleep 60 &** in **background**, bring it to foreground with **fg,** stop using **Ctrl + Z,** then resume in background using **bg**.



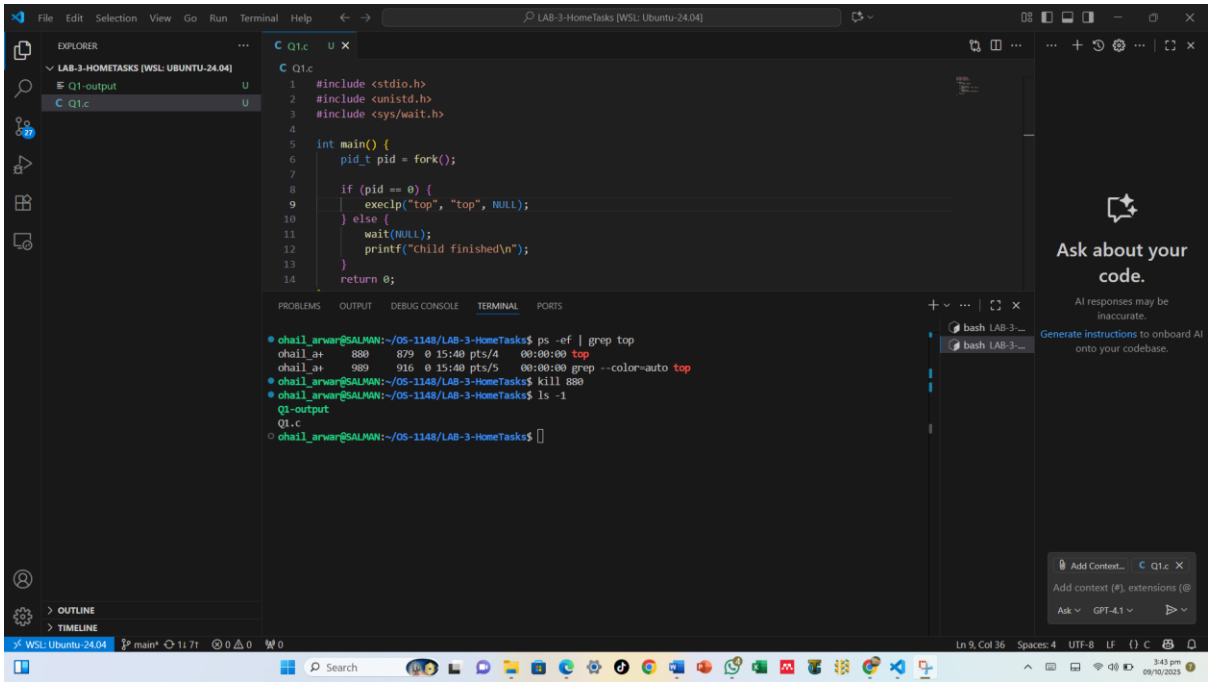
Part 6: C Programs on Processes

Program 1 – Exec with top

- Modify the exec program so that the child runs **top** instead of **ls-l.**
- Run the program.
- In another terminal, use **ps -ef | grep top** (or run **top**) to find the child's PID.
- Use the child's process ID to kill it manually.



Second Terminal:



Program 2 – Incomplete Program